CLAIMS

1. A method of retrieving object references in a stack comprising:

retrieving an object data structure from the stack;

extracting an object reference from one part of the data structure; and

extracting a reference to the next data structure in the stack from another part of the data structure.

2. A method as claimed in claim 1 further comprising retrieving the next data structure and retrieving the corresponding object reference and corresponding next data structure reference.

3. A method as claimed in claim 2 further comprising retrieving all linked data structures in the stack.

4. A method as claimed in claim 3 further comprising retrieving a last data structure having no next data structure reference.

5. A method as claimed in claim 4 further comprising retrieving the first object data structure in the stack referenced by a first object data structure pointer.

6. A method as claimed in claim 5 further comprising using the retrieved object references to define a root set of objects.

7. A method as claimed in claim 6 comprising:

defining a reachable set of objects as all objects referenced directly or indirectly by the root set objects.

8. A method as claimed in claim 7 further comprising identifying all objects within the process and reclaiming the memory space of all non-reachable objects.

9. A method as claimed in claim 7 or 8 further comprising moving reachable objects so that they are contiguous in memory and updating all object references in the stack by tracing through the chain of object data structures.

11. A method of managing a object in a stack based process comprising:

storing an object data structure in the stack comprising a reference to the object and a reference to a previously stored object data structure in the stack.

12. A method as claimed in claim 11 further comprising linking the object data structure to the previously stored object data structure.

13. A method as claimed in claim 12 further comprising:

storing a variable pointing to the previously stored object data structure at the top of the stack;

using the variable when storing a new object data structure; and updating the variable with the new object data structure reference.

14. A method as claimed in claim 13 further comprising:
saving the variable pointer;
storing the object data structure;
updating the variable with the reference to the latest stored object data structure;
performing the process ; and
restoring the stack pointer.

15. A method as claimed in claim 11 further comprising:
retrieving an object data structure and extracting the associated object reference and data structure reference;
using the associated data structure reference to retrieve the previously stored object data structure;
retrieving all the object references in the stack by tracing through the chained of object data structures.

16. A method as claimed in claim 15 further comprising using the retrieved object pointers to identify a root set of objects.

17. A method as claimed in claim 16 comprising:
identifying all objects referenced directly or indirectly by the root set objects and marking the root set and all referenced objects as reachable.

~~18. A method as claimed in claim 17 further comprising identifying all objects within the process and reclaiming the memory space of all non-reachable objects.~~

~~19. A method as claimed in claim 18 further comprising moving reachable objects in process memory so that they are contiguous and updating all object references in the stack by tracing through the chain of object data structures.~~

~~20. A system for retrieving object references in a stack comprising:~~

~~means for retrieving an object data structure from the stack;~~

~~means for extracting an object reference from one part of the data structure; and~~

~~means for extracting a reference to the next data structure in the stack from another part of the data structure.~~